

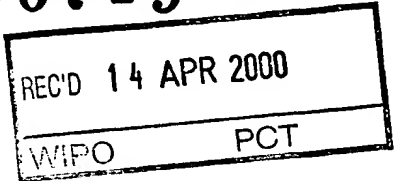
PCT/JP 00/00584

09/890749 04.04.00

日 本 国 特 許 庁

PATENT OFFICE
JAPANESE GOVERNMENT

2000/084



別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日

Date of Application:

1999年 2月 3日

出 願 番 号

Application Number:

平成11年特許願第025674号

出 願 人

Applicant (s):

日本電気株式会社

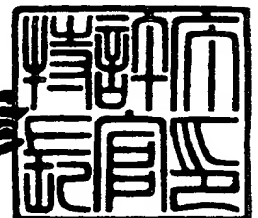
PRIORITY
DOCUMENT

SUBMITTED OR TRANSMITTED IN
COMPLIANCE WITH RULE 17.1(a) OR (b)

2000年 3月17日

特許庁長官
Commissioner,
Patent Office

近藤 隆彦



出証番号 出証特2000-3016395

【書類名】 特許願

【整理番号】 33509432

【提出日】 平成11年 2月 3日

【あて先】 特許庁長官殿

【国際特許分類】 G06F 7/00

【発明者】

 【住所又は居所】 東京都港区芝五丁目 7 番 1 号 日本電気株式会社内

 【氏名】 久村 孝寛

【特許出願人】

 【識別番号】 000004237

 【氏名又は名称】 日本電気株式会社

【代理人】

 【識別番号】 100088812

 【弁理士】

 【氏名又は名称】 ▲柳▼川 信

【手数料の表示】

 【予納台帳番号】 030982

 【納付金額】 21,000円

【提出物件の目録】

 【物件名】 明細書 1

 【物件名】 図面 1

 【物件名】 要約書 1

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 信号処理プロセッサ及びそれに用いる丸め機能付き積和演算器

【特許請求の範囲】

【請求項 1】 加数と被乗数と乗数との積和演算を行う積和演算器からなる丸め機能付き積和演算器を含む信号処理プロセッサであって、

前記丸め機能付き積和演算器に接続されたレジスタを有し、

前記丸め機能付き積和演算器は、前記レジスタ内の異なる位置から入力される加数を選択的に入力する選択入力手段と、前記選択入力手段で選択的に入力される加数に基づいた前記積和演算器の積和演算結果に対してデータ幅の大きなデータをデータ幅の小さなデータへ変換する丸め処理を行う丸め処理手段と、前記丸め処理手段で丸め処理された前記積和演算結果を前記レジスタ内の異なる位置に選択的に出力する選択出力手段とを含むことを特徴とする信号処理プロセッサ。

【請求項 2】 前記レジスタは、外部から読込んだデータを保持するための複数のレジスタからなることを特徴とする請求項 1 記載の信号処理プロセッサ。

【請求項 3】 前記選択入力手段で選択的に入力される加数に対してデータ幅の小さなデータをデータ幅の大きなデータへ変換する拡張処理を行って前記積和演算器へ入力する拡張処理手段を含むことを特徴とする請求項 1 または請求項 2 記載の信号処理プロセッサ。

【請求項 4】 前記丸め処理手段は、外部からの丸め指示に応じて前記丸め処理を実行するよう構成したことを特徴とする請求項 1 から請求項 3 のいずれか記載の信号処理プロセッサ。

【請求項 5】 前記選択入力手段は、前記レジスタから入力するデータの前記レジスタ内での位置を外部からの位置指示に応じて決定し、

前記選択出力手段は、前記レジスタへ出力するデータの前記レジスタ内での位置を前記位置指示に応じて決定するよう構成したことを特徴とする請求項 1 から請求項 4 のいずれか記載の信号処理プロセッサ。

【請求項 6】 加数と被乗数と乗数との積和演算を行う積和演算器を含む丸め機能付き積和演算器であって、外部に接続されたレジスタ内の異なる位置から

入力される加数を選択的に入力する選択入力手段と、前記選択入力手段で選択的に入力される加数に基づいた前記積和演算器の積和演算結果に対してデータ幅の大きなデータをデータ幅の小さなデータへ変換する丸め処理を行う丸め処理手段と、前記丸め処理手段で丸め処理された前記積和演算結果を前記レジスタ内の異なる位置に選択的に出力する選択出力手段とを有することを特徴とする丸め機能付き積和演算器。

【請求項 7】 前記レジスタは、外部から読込んだデータを保持するための複数のレジスタからなることを特徴とする請求項 6 記載の丸め機能付き積和演算器。

【請求項 8】 前記選択入力手段で選択的に入力される加数に対してデータ幅の小さなデータをデータ幅の大きなデータへ変換する拡張処理を行って前記積和演算器へ入力する拡張処理手段を含むことを特徴とする請求項 6 または請求項 7 記載の丸め機能付き積和演算器。

【請求項 9】 前記丸め処理手段は、外部からの丸め指示に応じて前記丸め処理を実行するよう構成したことを特徴とする請求項 6 から請求項 8 のいずれか記載の丸め機能付き積和演算器。

【請求項 10】 前記選択入力手段は、前記レジスタから入力するデータの前記レジスタ内での位置を外部からの位置指示に応じて決定し、

前記選択出力手段は、前記レジスタへ出力するデータの前記レジスタ内での位置を前記位置指示に応じて決定するよう構成したことを特徴とする請求項 6 から請求項 9 のいずれか記載の丸め機能付き積和演算器。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は信号処理プロセッサ及びそれに用いる丸め機能付き積和演算器に関し、特に信号処理プロセッサ等で使用する 16 ビットデータの効率の良い扱いを考慮した丸めつき積和演算回路に関する。

【0002】

【従来の技術】

信号処理プロセッサはメモリからデータを読み込んで、加減算、論理演算、乗算といったさまざまな演算を行う。特に、画像処理や音声処理等の信号処理プログラム中に頻繁に現れる積和演算を1サイクルで実行することができる積和演算器を搭載することによって、信号処理プロセッサの処理性能を飛躍的に向上させている。

【0003】

図6に従来の信号処理プロセッサにおける実行ユニット、レジスタ、メモリの構成例を示す。図6は「IEEE VLSI SIGNAL PROCESSING, VI」(pp. 93-101 1993) (以下、文献1とする)において紹介された信号処理プロセッサを示している。

【0004】

図6を参照すると、従来の信号処理プロセッサは40ビット幅のレジスタを8個持ち (以下、レジスタ50とする)、積和演算器52と、MUX (マルチプレクサ) 53と、ALU (Arithmetic and Logic Unit : 算術及び論理演算器) 54と、BSFT (バレルシフトユニット) 55と、Xメモリ57xと、Yメモリ57yとを備えている。以下、Xメモリ57x及びYメモリ57yをメモリ57x, 57yとする。

【0005】

メモリ57x, 57yとレジスタ50とはデータバス58x, 58yによって接続されている。積和演算器52とALU54とMUX53とBSFT55とはレジスタ50からの出力51a~51cによって接続されている。

【0006】

積和演算器52は積和演算を実行する。ALU54はMUX53によって選択された即値imm. またはレジスタ50の値を使用して算術または論理演算を実行する。BSFT55はMUX53によって選択された即値imm. またはレジスタ50の値を使用して算術または論理シフトを実行する。

【0007】

信号処理プログラムに頻繁に現れる積和演算とは、

$$A = A + B \times C \quad \dots\dots (1)$$

という乗算と加算とを行う演算である。

【0 0 0 8】

被乗数Bと乗数Cとの積を(1)式の右辺の加数Aに加算し、この結果を(1)式の左辺Aとする。(1)式の右辺の加数Aは多くの場合、繰返し実行される積和演算の結果であるが、メモリから読込むこともある。また、式(1)の右辺の「+」記号を「-」記号に変えた演算も積和演算である。

【0 0 0 9】

固定小数点の数値データを扱う一般的な積和演算器においては実用性や経済性等の点から、(1)式の右辺の被乗数B、乗数Cを16ビット幅のデータとすることが多い。この場合、被乗数Bと乗数Cとの積は最大で32ビット幅のデータとなるので、(1)式の右辺及び左辺Aは32ビット以上のデータ幅が必要である。

【0 0 1 0】

以上の理由から、積和演算の結果を保存するために、一般的な信号処理プロセッサでは32ビット以上の幅を持つレジスタを搭載している。このようなプロセッサでレジスタに16ビットデータを保持する時、レジスタの15～0ビットあるいは31～16ビットのどちらかに置くことになる。

【0 0 1 1】

図6に示した従来の信号処理プロセッサで(1)式の積和演算を実行する場合について、図7を参照して説明する。図7は図6に示した信号処理プロセッサのレジスタ50と積和演算器52とに注目して積和演算を実行する過程を示す図である。

【0 0 1 2】

まず、プロセッサの外部に接続されたメモリから(1)式の右辺の被乗数Bをレジスタ502に、乗数Cをレジスタ503に、加数Aをレジスタ501にそれぞれ読込む。

【0 0 1 3】

被乗数B及び乗数Cはレジスタ502、503の31～16ビットまたは15～0ビットのどちらにあってもよいが、ここでは、被乗数Bがレジスタ502の

3 1 ~ 1 6 ビットに、乗数 C がレジスタ 5 0 3 の 1 5 ~ 0 ビットにあるものとする。加数 A はレジスタの全てのビットを使用する。図 7 において、レジスタ 5 0 1 ~ 5 0 3 の下に書かれた数字はビット位置を示す。

【 0 0 1 4 】

続いて、加数 A を積和演算器の ACC (アキュムレータ) 5 2 3 に格納する。さらに、被乗数 B と乗数 C とを積和演算器 5 2 内にある乗算器 5 2 1 に入力し、これらの積を計算する。被乗数 B と乗数 C との積は加減算器 (±) 5 2 2 によって、ACC 5 2 3 内に格納した加数 A に加算される。最後に、加算結果は ACC 5 2 3 に一旦格納され、積和演算器の出力 5 6 を通じて加数 A が格納されていたレジスタ 5 0 1 に書戻される。

【 0 0 1 5 】

図 6 のような構成の信号処理プロセッサ上で、1 6 ビットデータの加数をメモリから読み込み、これに対してある積和演算を行って、その結果を再び 1 6 ビットデータとしてメモリに保存する処理を考える。つまり、途中の演算結果は別にしておき、入出力データの幅を 1 6 ビットに統一した積和演算である。

【 0 0 1 6 】

これはある積和演算の被乗数または乗数を別の積和演算の加数として使うという処理を繰返し行う場合に発生する。このような処理では加数、被乗数、乗数が全て 1 6 ビットデータとして扱われるので、これらの値によっては演算結果がオーバーフローする可能性がある。しかしながら、値の範囲を適切に選ぶことが可能な場合にはオーバーフローを起こさずに、演算を実行することが可能である。

【 0 0 1 7 】

図 6 に示した従来の信号処理プロセッサで入出力データの幅を 1 6 ビットに統一した積和演算を実行する場合について、図 8 を参照して説明する。図 8 では図 7 と同様に、プロセッサの外部に接続されたメモリから被乗数 B をレジスタ 5 0 2 の 3 1 ~ 1 6 ビットに、乗数 C をレジスタ 5 0 3 の 1 5 ~ 0 ビットに、加数 A をレジスタ 5 0 1 の 3 1 ~ 1 6 ビットにそれぞれ読み込む。

【 0 0 1 8 】

ここで、1 6 ビットデータである加数 A をレジスタ 5 0 1 に読み込む際に、A の

符号が39～32ビットに挿入され、15～0ビットには“0”が挿入される。
レジスタ501～503にデータが格納された状態を50nとする。

【0019】

次の状態50n1では積和演算が実行され、 $A+B \times C$ がレジスタ501に格納される。また、状態50n2では積和演算の結果40ビットをALU54によって16ビットに丸め、丸めた結果はレジスタ501に格納される。最後に、丸めたデータをメモリに格納する。

【0020】

以上の処理においては二つの問題がある。一つ目の問題はメモリから読込んだ加数Aのデータ幅と積和演算器52が要求する加数のデータ幅との相違である。加数Aは16ビット幅のデータであるから、積和演算を行うためには40ビットのデータ幅に拡張しなければならない。したがって、同一レジスタ内に二つの16ビット加数を置くことができない。

【0021】

二つ目の問題は積和演算器の演算結果のデータ幅とメモリ格納時のデータ幅との相違である。従来の信号処理プロセッサが備える積和演算器は40ビットのデータ幅をもつ演算結果を出力するので、16ビット幅のデータとしてメモリに格納する場合、40ビットを16ビットに丸める必要がある。つまり、積和演算に加えて、丸め処理を行わなければならない。

【0022】

さらに、図6に示す従来の信号処理プロセッサの性能向上を図るために、メモリとレジスタとの間のバス幅を32ビットに増やした場合を考えると、一つのデータバスから16ビットデータを同時に二つ読込むことができる。

【0023】

このような信号処理プロセッサにおいて、再び入出力データの幅を16ビットに統一した積和演算について考えると、被乗数及び乗数は16ビットデータであるから、32ビットの転送能力を活かしてそれぞれを二つずつ同時に読込むことができる。読込まれた二つのデータはレジスタの31～16ビットと15～0ビットとに格納されるものとする。

【 0 0 2 4 】

同様に、加数についても 3 2 ビット転送によって二つのデータを同時にレジスタに読み込むことが可能である。しかしながら、これはうまく機能しない。なぜならば、1 6 ビットデータの加数はレジスタの 3 1 ~ 1 6 ビットにあって、1 5 ~ 0 ビットは“0”でなければならないからである。

【 0 0 2 5 】

つまり、図 8 において、加数 A が格納されているレジスタ 5 0 1 の 1 5 ~ 0 ビットに別の加数がある場合、そのままでは正しい演算を実行することができないのである。したがって、加数は一つずつレジスタに読み込まなければならない。

【 0 0 2 6 】

3 2 ビット転送を使用して加数を同一のレジスタに二つ読み込んだ場合には、レジスタ・レジスタ間転送またはシフト等を用いて、別のレジスタに一つずつデータを格納しなければならない。メモリ・レジスタ間の転送命令は半分にすることができても、レジスタ・レジスタ間の転送が必要となるので、この場合、メモリ・レジスタ間の 3 2 ビット転送能力が活かされないことになる。

【 0 0 2 7 】

【発明が解決しようとする課題】

上述した従来の信号処理プロセッサでは、上記のように、1 6 ビットデータの扱いに問題を抱えている。一つは、入出力データの幅を 1 6 ビットに統一した積和演算において、余分な資源を消費してしまうことである。加数の 1 6 ビットデータは積和演算器の要求するデータ幅に合わせるためにレジスタの幅に拡張して格納しなければならない。

【 0 0 2 8 】

さらに、積和演算の結果はレジスタの幅と同じデータ幅になっているので、1 6 ビットのデータにするためには丸め処理が必要である。さらに、この問題によって、メモリ・レジスタ間の転送効率向上が期待できないという問題が引き起こされる。

【 0 0 2 9 】

例えば、メモリ・レジスタ間の転送能力を 1 6 ビットの倍の 3 2 ビット幅に拡張

大したとしても、16ビットの加数データを複数個同時にレジスタに読み込み、効率よく演算するためにはレジスタ・レジスタ間での加数データの転送が不可欠である。したがって、演算を実行するまでに必要なメモリ・レジスタ間のデータ転送において、効率の向上は期待できない。

【0030】

そこで、本発明の目的は上記の問題点を解消し、16ビットデータの効率の良い扱いを考慮した信号処理プロセッサ及びそれに用いる丸め機能付き積和演算器を提供することにある。さらに詳しく言うと、レジスタの中での加数の位置を考慮した16ビットの積和演算を実行可能な信号処理プロセッサ及びそれに用いる丸め機能付き積和演算器を提供することにある。

【0031】

【課題を解決するための手段】

本発明による信号処理プロセッサは、加数と被乗数と乗数との積和演算を行う積和演算器からなる丸め機能付き積和演算器を含む信号処理プロセッサであって、前記丸め機能付き積和演算器に接続されたレジスタを備え、

前記丸め機能付き積和演算器は、前記レジスタ内の異なる位置から入力される加数を選択的に入力する選択入力手段と、前記選択入力手段で選択的に入力される加数に基づいた前記積和演算器の積和演算結果に対してデータ幅の大きなデータをデータ幅の小さなデータへ変換する丸め処理を行う丸め処理手段と、前記丸め処理手段で丸め処理された前記積和演算結果を前記レジスタ内の異なる位置に選択的に出力する選択出力手段とを具備している。

【0032】

本発明による丸め機能付き積和演算器は、加数と被乗数と乗数との積和演算を行う積和演算器からなる丸め機能付き積和演算器であって、外部に接続されたレジスタ内の異なる位置から入力される加数を選択的に入力する選択入力手段と、前記選択入力手段で選択的に入力される加数に基づいた前記積和演算器の積和演算結果に対してデータ幅の大きなデータをデータ幅の小さなデータへ変換する丸め処理を行う丸め処理手段と、前記丸め処理手段で丸め処理された前記積和演算結果を前記レジスタ内の異なる位置に選択的に出力する選択出力手段とを備えて

いる。

【0033】

すなわち、本発明の丸め機能付き積和演算器は、選択入力及び拡張手段と、丸め及び選択出力手段と、積和演算器とを備える。丸め機能付き積和演算器は制御信号Round, Positionによってその動作を決定する。制御信号Roundが“0”の場合、本発明の丸め機能付き積和演算器は従来の積和演算器として動作する。

【0034】

制御信号Roundが“1”の場合、本発明の丸め機能付き積和演算器の動作は制御信号Positionによって異なる。選択入力及び拡張手段は外部に接続されたレジスタにおける16ビットデータである加数の位置を制御信号Positionで与えることによって、加数を積和演算器が要求するデータ幅に拡張する。

【0035】

丸め及び選択出力手段は積和演算の結果として得られるレジスタ幅のデータを16ビットに丸め処理した後、制御信号Positionが示すレジスタにおける加数の位置に丸め処理したデータを出力する。

【0036】

このような構成によって、丸め機能付き積和演算器の外部に接続された32ビット以上の幅をもつレジスタ内の31～16ビットあるいは15～0ビットにある16ビット幅の加数に対して、互いに影響を与えずに丸め付き積和演算を実行することが可能となる。

【0037】

【発明の実施の形態】

次に、本発明の実施の形態について図面を参照して説明する。図1は本発明の実施の形態の構成を示すブロック図である。同図において、丸め機能付き積和演算器4は16ビットデータの効率の良い扱いを考慮した丸め機能付き積和演算器である。さらに詳しく言うと、32ビット以上の幅をもつレジスタ1内に二つの別々の16ビットデータがある場合に、互いのデータに影響を与えないような積

和演算が実行可能な改良された丸め機能付き積和演算器である。

【0038】

丸め機能付き積和演算器4は外部にレジスタを接続して動作させる。丸め機能付き積和演算器4は外部のレジスタから加数、被乗数、乗数の三つのデータを取込む。また、丸め機能付き積和演算器4は演算の結果を書出す出力を三つ備えている。さらに、丸め機能付き積和演算器4は演算の動作を決定する制御信号を二つ持つ。

【0039】

加数は丸め機能付き積和演算器4の外部に接続された40ビットレジスタ1の31～16ビットあるいは15～0ビットに格納されている。被乗数及び乗数は外部のレジスタに格納されており、丸め機能付き積和演算器4に入力される。

【0040】

図1では加数が格納されている40ビットレジスタ1のみを示し、被乗数や乗数が格納されているレジスタは示していない。40ビットレジスタ1に格納されたデータは15～0ビットが入力データ46L、31～16ビットが入力データ46H、39～32ビットが入力データ46Eとして丸め機能付き積和演算器4に入力される。また、被乗数は16ビットデータ45、乗数は16ビットデータ44として丸め機能付き積和演算器4へ入力される。

【0041】

加数と被乗数と乗数との積和演算及び丸め処理の結果は出力47L、47H、47Eとして40ビットレジスタ1へ書戻される。出力47L、47Hは16ビットデータであり、出力47Eは8ビットデータである。出力47Lは40ビットレジスタ1の15～0ビットへ、出力47Hは40ビットレジスタ1の31～16ビットへ、出力47Eは40ビットレジスタ1の39～32ビットへそれぞれ出力される。

【0042】

丸め機能付き積和演算器4の動作を制御するのは二つの制御信号Round、Positionである。制御信号Positionは40ビットレジスタ1における加数の位置を示す信号である。加数が40ビットレジスタ1の31～16

ビットにある場合に制御信号 $Position=1$ 、15～0ビットにある場合に制御信号 $Position=0$ とする。

【0043】

制御信号 $Round$ は丸め機能付き積和演算器 4 において丸め処理を行うかどうかを決める信号である。ここで、丸め処理とはデータ幅の大きなデータをデータ幅の小さなデータへ変換する操作である。この逆の操作、つまりデータ幅の小さなデータをデータ幅の大きなデータへ変換する操作を拡張処理と呼ぶことにする。丸め処理を行う場合は制御信号 $Round=1$ 、丸め処理を行わない場合は制御信号 $Round=0$ とする。

【0044】

さらに、図 1 の丸め機能付き積和演算器 4 の内部構成について説明する。丸め機能付き積和演算器 4 は積和演算器 41 と、選択入力及び拡張手段 42 と、丸め及び選択出力手段 43 とからなる。

【0045】

選択入力及び拡張手段 42 はレジスタ 1 からの入力データ 46E, 46H, 46L からデータを選択して拡張し、40ビットデータを出力する。また、選択入力及び拡張手段 42 は制御信号 $Position$, $Round$ によって制御される。積和演算器 41 は選択入力及び拡張手段 42 によって得られる加数と乗数 44 と被乗数 45 とによる積和演算を実行する。

【0046】

丸め及び選択出力手段 43 は積和演算器 41 の出力を40ビットから16ビットへ丸めて、40ビットレジスタ 1 の指定された位置へ出力する。また、丸め及び選択出力手段 43 は制御信号 $Position$, $Round$ によって制御される。

【0047】

本発明の丸め機能付き積和演算器 4 は従来の積和演算器 41 に、選択入力及び拡張手段 42 と丸め及び選択出力手段 43 とを加えることによって16ビットデータの扱いを改善したものである。

【0048】

次に、図 1 を参照して本発明の丸め機能付き積和演算器 4 の動作について説明する。選択入力及び拡張手段 4 2 は 40 ビットレジスタ 1 から入力される入力データ 4 6 E, 4 6 H, 4 6 L に基づいて 40 ビットデータ 4 8 を構成する。これは制御信号 Round, Position によって制御される。

【0049】

制御信号 Round が “0” の場合は丸め処理を行わないので、丸め機能付き積和演算器 4 は従来の積和演算器と同じ演算結果が得られるように動作しなければならない。この時、40 ビットレジスタ 1 から入力される入力データ 4 6 E, 4 6 H, 4 6 L はそのまま 40 ビットデータとして積和演算器 4 1 へ出力される。

【0050】

一方、制御信号 Round が “1” の場合は制御信号 Position によって動作が異なる。制御信号 Round が “1” で、制御信号 Position が “0” の場合には、加数が 40 ビットレジスタ 1 の 15～0 ビットにあるものとして処理を行う。この場合、40 ビットレジスタ 1 の 15～0 ビットにある加数 4 6 L を 40 ビットに拡張して積和演算器 4 1 へ出力する。

【0051】

制御信号 Round が “1” で、制御信号 Position が “1” の場合には、加数が 40 ビットレジスタ 1 の 31～16 ビットにあるものとして処理を行う。この場合、40 ビットレジスタ 1 の 31～16 ビットにある加数 4 6 H を 40 ビットに拡張して積和演算器 4 1 へ出力する。

【0052】

積和演算器 4 1 は 16 ビットデータ 4 4, 4 5 の積を計算し、これを 40 ビットデータ 4 8 に加算した結果を、同じく 40 ビットデータ 4 9 として出力する。最後に、丸め及び選択出力手段 4 3 は積和演算器 4 1 で計算された積和演算の結果である 40 ビットデータ 4 9 に丸め処理を行って、16 ビットデータとして 40 ビットレジスタ 1 へ出力する。ここで、丸め処理を行うかどうかは制御信号 Round によって決定する。丸めた 16 ビットデータを 40 ビットレジスタ 1 のどの部分に出力するかは制御信号 Position によって決定する。制御信号

R o u n d が “ 0 ” の場合には丸め処理を行わずに 4 0 ビットデータ 4 9 を出力 4 7 E, 4 7 H, 4 7 L にわけてレジスタ 1 へ出力する。

【 0 0 5 3 】

制御信号 R o u n d が “ 1 ” で、制御信号 P o s i t i o n が “ 0 ” の場合には加数が 4 0 ビットレジスタ 1 の 1 5 ～ 0 ビットにあるので、4 0 ビットデータ 4 9 に丸め処理を施して 1 6 ビットデータとし、これを出力 4 7 L として 4 0 ビットレジスタ 1 の 1 5 ～ 0 ビットに出力する。この時、出力 4 7 H, 4 7 E は動作しないので、4 0 ビットレジスタ 1 の 3 9 ～ 1 6 ビットは変化しない。

【 0 0 5 4 】

制御信号 R o u n d が “ 1 ” で、制御信号 P o s i t i o n が “ 1 ” の場合には加数が 4 0 ビットレジスタ 1 の 3 1 ～ 1 6 ビットにあるので、4 0 ビットデータ 4 9 に丸め処理を施して 1 6 ビットデータとし、これを出力 4 7 H として 4 0 ビットレジスタ 1 の 3 1 ～ 1 6 ビットに出力する。この時、出力 4 7 L, 4 7 E は動作しないので、4 0 ビットレジスタ 1 の 3 9 ～ 3 2 ビット及び 1 5 ～ 0 ビットは変化しない。

【 0 0 5 5 】

図 2 は図 1 の選択入力及び拡張手段 4 2 の動作を示す図である。この図 2 を参照して選択入力及び拡張手段 4 2 の動作について説明する。選択入力及び拡張手段 4 2 の目的は丸め処理の有無、積和演算に必要な加数の 4 0 ビットレジスタ 1 における位置を考慮し、4 0 ビットレジスタ 1 から 4 0 ビットデータを作成することである。

【 0 0 5 6 】

選択入力及び拡張手段 4 2 における丸め処理の有無は制御信号 R o u n d によって決定し、加数の 4 0 ビットレジスタ 1 における位置は制御信号 P o s i t i o n によって決定する。

【 0 0 5 7 】

例えば、制御信号 R o u n d が “ 0 ” である場合、制御信号 P o s i t i o n が何であろうと、4 0 ビットレジスタ 1 から入力された入力データ 4 6 E, 4 6 H, 4 6 L によって 4 0 ビットデータ 4 8 を構成して出力する。この場合、1 6

ビットデータ46Lは40ビットデータ48の15～0ビット、16ビットデータ46Hは同じく40ビットデータ48の31～16ビット、8ビットデータ46Eは同じく40ビットデータ48の39～32ビットとなる。

【0058】

一方、制御信号Roundが“1”である場合には、制御信号Positionによって動作が異なる。制御信号Roundが“1”でかつ制御信号Positionが“0”である場合、加数は40ビットレジスタ1の15～0ビットにある。そこで、40ビットレジスタ1の15～0ビットにある16ビットデータ46Lを40ビット幅のデータへ拡張し、40ビットデータ48とする。

【0059】

制御信号Roundが“1”でかつ制御信号Positionが“1”である場合、加数は40ビットレジスタ1の31～16ビットにある。そこで、40ビットレジスタ1の31～16ビットにある16ビットデータ46Hを40ビット幅のデータへ拡張し、40ビットデータ48とする。

【0060】

固定小数点データの場合、16ビットデータ46Lまたは16ビットデータ46Hから40ビットデータ48への拡張は次のようにして行う。まず、拡張する16ビットデータを40ビットデータ48の31～16ビットとする。

【0061】

さらに、拡張する16ビットデータの符号ビットを取出し、その符号ビットを40ビットデータ48の39～32ビットとする。通常、符号ビットは1ビットの情報であるから、40ビットデータ48の39～32ビットには符号ビットが繰返し挿入される。最後に、40ビットデータ48の15～0ビットに0を挿入する。

【0062】

図3は図1の丸め及び選択出力手段43の動作を示す図である。この図3を参照して丸め及び選択出力手段43の動作について説明する。丸め及び選択出力手段43へは積和演算器41から演算結果49が入力される。

【0063】

丸め及び選択出力手段 43 の目的は丸め処理の有無、積和演算に使用した加数の 40 ビットレジスタ 1 における位置を考慮し、入力 40 ビットデータ 49 を 40 ビットレジスタ 1 へ出力することである。

【0064】

丸め及び選択出力手段 43 における丸め処理の有無は制御信号 Round によって決定し、加数のレジスタ 1 における位置は制御信号 Position によって決定する。

【0065】

例えば、制御信号 Round が “0” である場合、制御信号 Position が何であろうと、丸め処理を行わずに入力 40 ビットデータ 49 をそのまま 40 ビットレジスタ 1 へ出力する。

【0066】

この時、16 ビット出力データ 47L は入力 40 ビットデータ 49 の 15～0 ビット、16 ビット出力データ 47H は同じく入力 40 ビットデータ 49 の 31～16 ビット、8 ビット出力データ 47E は同じく入力 40 ビットデータ 49 の 39～32 ビットとなる。

【0067】

一方、制御信号 Round が “1” である場合、制御信号 Position によって動作が異なる。制御信号 Round が “1” でかつ制御信号 Position が “0” である場合、加数は 40 ビットレジスタ 1 の 15～0 ビットにある。そこで、丸め及び選択出力手段 43 は入力 40 ビットデータ 49 に丸め処理を行って 16 ビットデータとし、これを出力 47L とする。これは 40 ビットレジスタ 1 の 15～0 ビットに書込まれる。この時、出力 47H 及び出力 47E は何も出力しない。

【0068】

制御信号 Round が “1” でかつ制御信号 Position が “1” である場合、加数は 40 ビットレジスタ 1 の 31～16 ビットにある。そこで、丸め及び選択出力手段 43 は入力 40 ビットデータ 49 に丸め処理を行って 16 ビットデータとし、これを出力 47H とする。これは 40 ビットレジスタ 1 の 31～1

6ビットに書込まれる。この時、出力47L及び出力47Eは何も出力しない。

【0069】

尚、本発明の実施の形態では40ビットレジスタ1のデータ幅が40ビットの場合について述べたが、そのデータ幅が32ビットの場合にも適用可能である。その場合、入力46Eと出力47Eとが不要となる。

【0070】

さらに、本発明の実施の形態ではレジスタ幅とデータサイズとを変えることで、Mビット幅のレジスタで、 $N \leq M/2$ なるNビットのデータを扱うように修正することも可能である。この場合、図1において、1はMビット幅のレジスタ、2, 3, 44, 45, 46H, 46L, 47H, 47LはNビット幅のデータ、46E, 47Eは $M-2N$ ビット幅のデータ、48, 49はMビット幅のデータとなる。

【0071】

積和演算器41はNビットデータ44, 45の積を計算し、選択入力及び拡張手段42から受取ったMビットデータ48に加算する。その演算結果49は丸め及び選択出力手段43によって丸め処理され、Nビットデータとしてレジスタ1に書戻される。選択入力及び拡張手段42及び丸め及び選択出力手段43の動作はそれぞれ図2及び図3において、ビット位置を示す数字39を $M-1$ に、同じく32を $2N$ に、31を $2N-1$ に、16を N に、15を $N-1$ に変更したものとなる。但し、 $M=2N$ の場合には入力46Eと出力47Eとが不要となる。

【0072】

図4は本発明の一実施例による丸め機能付き積和演算器の構成を示すブロック図である。図においては外部に40ビットレジスタ61~63を接続し、かつMUX（マルチプレクサ）64, 65によってレジスタ61~63の31~16ビットあるいは15~0ビットのどちらかを選択して丸め機能付き積和演算器4に入力できるようにした構成例を示している。この図4において、レジスタ61~63はメモリ（図示せず）からデータを読込んだり、メモリへデータを保存することができるものとする。

【0073】

図5は図4の丸め機能付き積和演算器4を使用して加数、被乗数、乗数の全てが16ビットデータである積和演算を実行した場合のレジスタの状態を示す図である。これら図4及び図5を参照して丸め機能付き積和演算器4の動作について説明する。

【0074】

レジスタ61には「A」及び「D」の16ビットデータがメモリから読込まれた状態にある。レジスタ61の31～16ビットには加数Aが、15～0ビットには別の加数Dが格納されている。

【0075】

この状態で、まず $A = A + B \times C$ という積和演算を実行する。ここで、丸め機能付き積和演算器4への制御信号Round=1、制御信号Position=1とする。

【0076】

丸め機能付き積和演算器4はレジスタ61から加数Aを取込み、レジスタ62からMUX64によって選択された被乗数Bと、レジスタ63からMUX65によって選択された乗数Cとによって積和演算と丸め処理とを行う。その結果は再びレジスタ61に書戻される。この時、レジスタ61の31～16ビットに演算結果が書込まれ、レジスタ61の15～0ビットは変化しない。この状態を6n1とする。

【0077】

次に、 $D = D + E \times F$ という積和演算を実行する。ここで、丸め機能付き積和演算器4への制御信号Round=1、制御信号Position=0とする。丸め機能付き積和演算器4はレジスタ61から加数Dを取込み、レジスタ62からMUX64によって選択された被乗数Eと、レジスタ63からMUX65によって選択された乗数Fとによって積和演算と丸め処理とを行う。その結果は再びレジスタ61に書戻される。この時、レジスタ61の15～0ビットに演算結果が書込まれ、レジスタ61の31～16ビットは変化しない。この状態を6n2とする。

【0078】

状態 6 n 2 のレジスタ 6 1 には $A + B \times C$ と $D + E \times F$ との演算結果が 1 6 ビットに丸めて 3 1 ~ 1 6 ビットと 1 5 ~ 0 ビットとにそれぞれ格納されており、そのままメモリへ保存することが可能となる。

【0079】

図 4 ではレジスタ 6 1 は 4 6 L, 4 6 H, 4 6 E, 4 7 E, 4 7 H, 4 7 L と、レジスタ 6 2 は MUX 6 4 を介して被乗数 1 6 ビットデータ 4 5 と、レジスタ 6 3 も MUX 6 5 を介して乗数 1 6 ビットデータ 4 4 と固定的に接続されているが、これらの組合せを自由に選択可能なように修正することは容易である。

【0080】

これは本発明の一実施例による丸め機能付き積和演算器 4 を、複数のレジスタをもつ信号処理プロセッサで使用する場合に特に有効である。もちろん、信号処理プロセッサに含まれない回路として本発明の一実施例による丸め機能付き積和演算器 4 を使用することも可能である。

【0081】

本発明の一実施例による丸め機能付き積和演算器 4 を使用すれば、図 5 に示したように、レジスタ内の 3 1 ~ 1 6 ビットあるいは 1 5 ~ 0 ビットにある 1 6 ビット幅の加数に対して、他のビットに影響を与えずに積和演算を実行することが可能となる。この効果によって、レジスタ内に異なる 1 6 ビット幅の加数を混在させることが可能となる。この効果は積和演算に使用するレジスタ数の削減につながる。

【0082】

さらに、本発明の一実施例による丸め機能付き積和演算器 4 を 3 2 ビット幅以上のレジスタをもつ信号処理プロセッサに搭載した場合、2 つの 1 6 ビット幅の加数をメモリから同時に読込み、一つのレジスタ内に混在させても正しく積和演算を実行することができるため、レジスタ・メモリ間の 3 2 ビット転送を活かすことができる。

【0083】

以上のことを、LMS (Least Mean Square) 適応フィルタの実装を例にとって示す。LMS 適応フィルタは時刻 i における入力信号を x_i

とするタップ数 T のFIR (Finite Impulse Response) フィルタ (係数 w_j) ($j=0, 1, \dots, T-1$) の出力 y_i と所望信号 d_i との誤差 e_i によって、FIRフィルタの係数を勾配法を用いて適応的に更新するものである。 y_i の計算と w_j の更新式とは、

$$y_i = \sum x_{i-j} w_j \quad \dots\dots (2)$$

$$e_i = d_i - y_i \quad \dots\dots (3)$$

$$w_j = w_j + \mu e_i x_{i-j} \quad \dots\dots (4)$$

という式で表される。但し、 \sum は $j=0$ から $j=T-1$ の総和であり、 μ は非常に小さな正の定数であり、 $j=0, 1, \dots, T-1$ である。

【0084】

(4) 式を参照すると、 μe_i は一度計算しておけば全ての係数について同じ値を使用することができる。そのため、(4) 式のフィルタ係数の更新において係数毎に必要な基本操作は、

(1) メモリから x_{i-j} を読み込む

(2) メモリから w_j を読み込む

(3) 加数 w_j 、被乗数 μe_i 、乗数 x_{i-j} による積和演算

$$w_j = w_j + \mu e_i x_{i-j}$$

(4) w_j をメモリへ保存

という四つの処理であることがわかる。

【0085】

信号処理プロセッサでこれら四つの処理を行う場合、 x_{i-j} 、 w_j 、 e_i は16ビットデータ、 y_i は32ビット以上のデータとする場合が多い。これは信号処理プロセッサに搭載された積和演算器の仕様によるものである。 x_{i-j} 、 w_j 、 e_i 、 y_i を上記のようなデータ幅とした時、 w_j が加数となる(3)の積和演算はデータ幅を16ビットに統一した演算になる。

【0086】

上記の基本操作において、(1)と(2)と(4)とのメモリ・レジスタ間の転送は16ビットデータの転送であるから、32ビット幅のデータバスをもつ信号処理プロセッサならば、二つのイタレーションの転送をまとめて一回で行うこ

とができる。

【0087】

(2) の w_j の読込みに 32 ビット転送を使用した場合、読込まれた二つのデータはレジスタの 31~16 ビットと 15~0 ビットとに格納される。従来の積和演算器を搭載するプロセッサでは、このような配置で格納された w_j をそのままの状態で作算することは不可能だが、本発明の一実施例による丸め機能付き積和演算器 4 を搭載した信号処理プロセッサならば、すぐに積和演算を実行することができる。その結果、 w_j をメモリへ保存する (4) の処理においても、演算後すぐに 32 ビット転送を使用することができる。

【0088】

以上のようにして、32 ビット幅のデータバスをもつ信号処理プロセッサで 16 ビットデータの積和演算を実行する場合でも、本発明一実施例による丸め機能付き積和演算器 4 を使用することによって、メモリ・レジスタ間の 32 ビット転送を活かすことができる。

【0089】

【発明の効果】

以上説明したように本発明によれば、加数と被乗数と乗数との積和演算を行う積和演算器からなる丸め機能付き積和演算器において、外部に接続されたレジスタ内の異なる位置から入力される加数を選択的に入力し、選択的に入力される加数に基づいた積和演算器の積和演算結果に対してデータ幅の大きなデータをデータ幅の小さなデータへ変換する丸め処理を行い、その丸め処理された積和演算結果をレジスタ内の異なる位置に選択的に出力することによって、レジスタの中での加数の位置を考慮した 16 ビットの積和演算を実行することができるという効果がある。

【図面の簡単な説明】

【図 1】

本発明の実施の形態の構成を示すブロック図である。

【図 2】

図 1 の選択入力及び拡張手段の動作を示す図である。

【図 3】

図 1 の丸め及び選択出力手段の動作を示す図である。

【図 4】

本発明の一実施例による丸め機能付き積和演算器の構成を示すブロック図である。

【図 5】

図 4 の丸め機能付き積和演算器を使用して加数、被乗数、乗数の全てが 1 6 ビットデータである積和演算を実行した場合のレジスタの状態を示す図である。

【図 6】

従来の信号プロセッサの構成例を示すブロック図である。

【図 7】

従来の信号プロセッサによる加数が 4 0 ビットデータで、被乗数及び乗数が 1 6 ビットデータである積和演算の実行例を示すブロック図である。

【図 8】

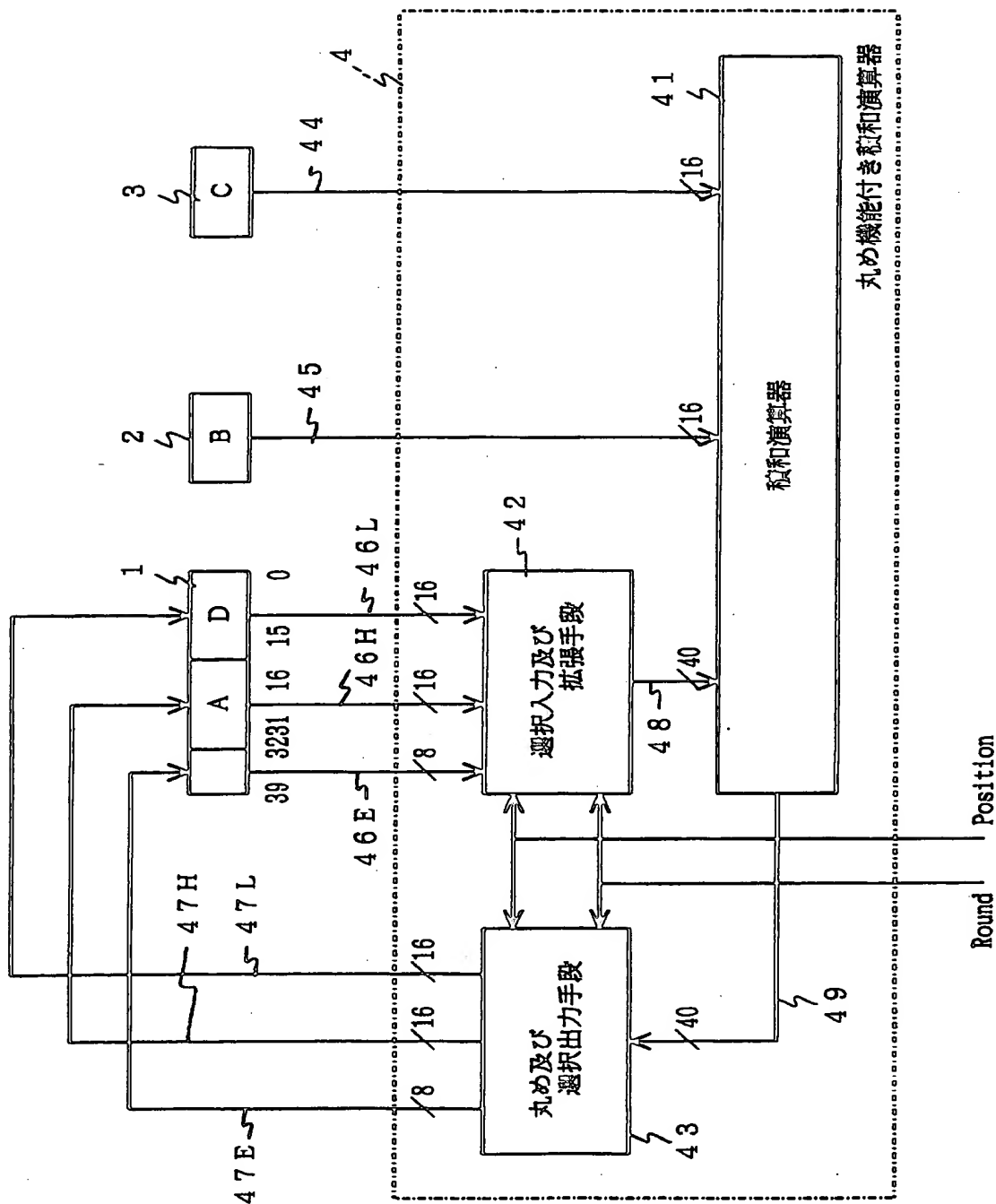
従来の信号プロセッサによる加数、被乗数、乗数の全てが 1 6 ビットデータである積和演算を実行した場合のレジスタの状態を示す図である。

【符号の説明】

- 1, 6 1 ~ 6 3 4 0 ビットレジスタ
- 2, 3 1 6 ビットデータ
- 4 丸め機能付き積和演算器
- 4 1 積和演算器
- 4 2 選択入力及び拡張手段
- 4 3 丸め及び選択出力手段
- 6 4, 6 5 マルチプレクサ
- A 加数
- B 被乗数
- C 乗数

【書類名】 図面

【図 1】



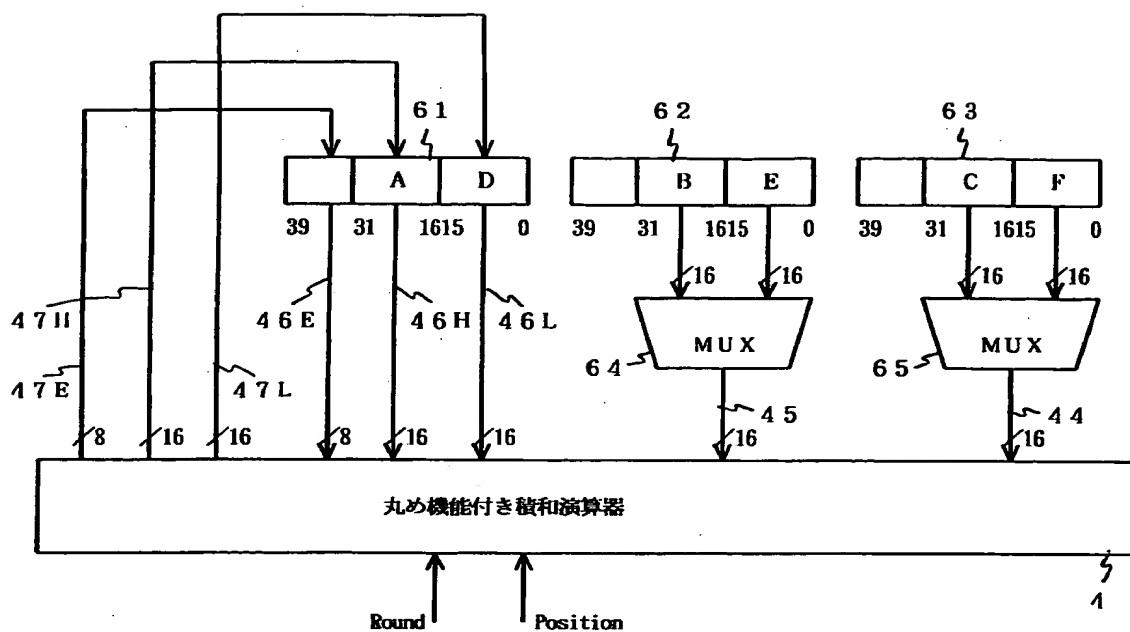
【図 2】

制御信号		出力
Round	Position	
0	0	レジスタ 1 の 3 9 ~ 0 ビットをそのまま出力する
0	1	
1	0	レジスタ 1 の 1 5 ~ 0 ビットを 4 0 ビットデータへ拡張して出力する
1	1	レジスタ 1 の 3 1 ~ 1 6 ビットを 4 0 ビットデータへ拡張して出力する

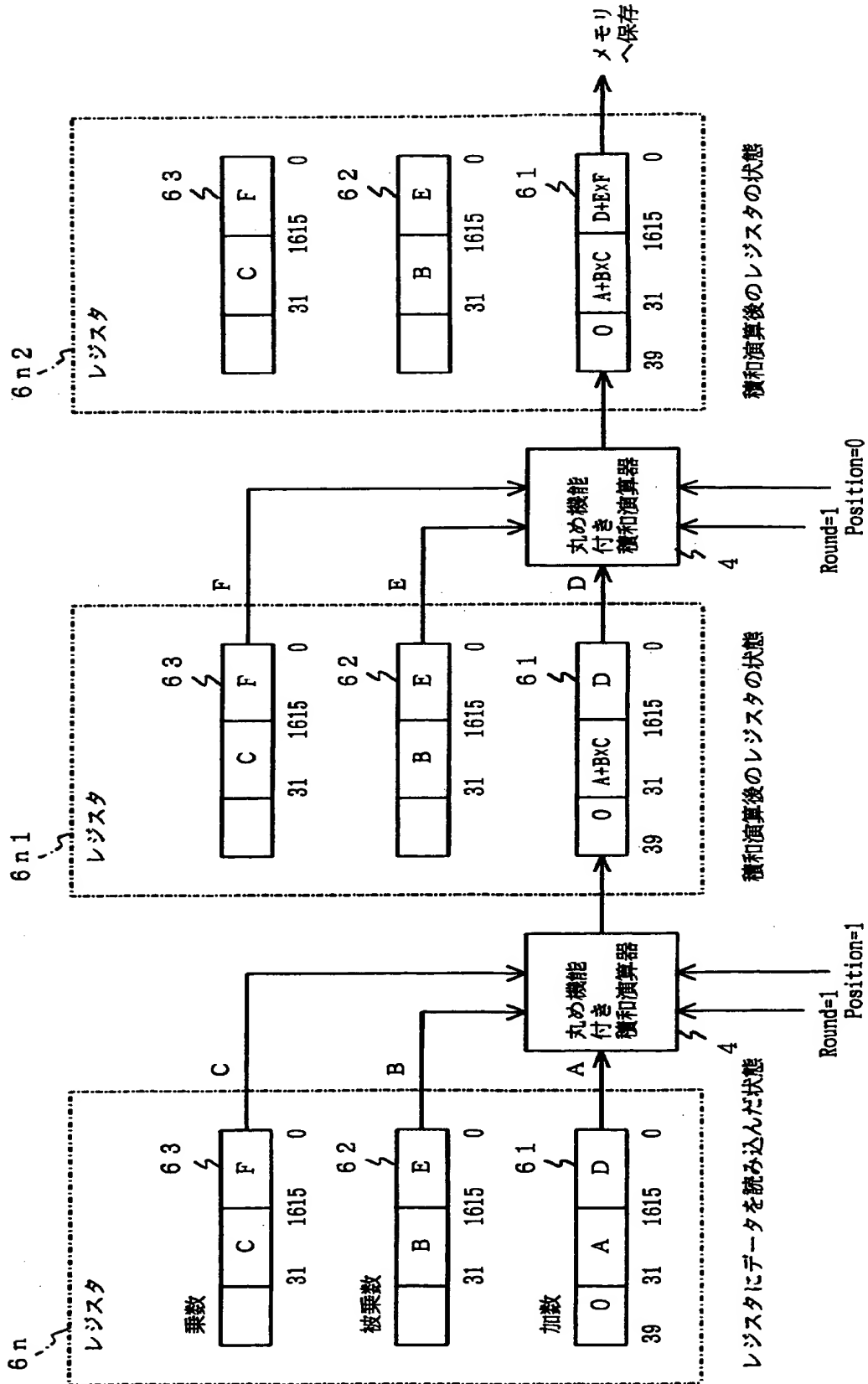
【図 3】

制御信号		出力			備考
Round	Position	出力 4 7 E	出力 4 7 H	出力 4 7 L	
0	0	入力データ 49 の 3 9 ~ 3 2 ビット	入力データ 49 の 3 1 ~ 1 6 ビット	入力データ 49 の 1 5 ~ 0 ビット	入力データ 49 を丸め処理をせずそのままレジスタ 1 へ出力
0	1				
1	0	何も出力しない	何も出力しない	入力データ 49 を丸め処理した 1 6 ビットデータ	入力データ 49 を丸め処理してレジスタ 1 1 5 ~ 0 ビットへ出力
1	1	何も出力しない	入力データ 49 を丸め処理した 1 6 ビットデータ	何も出力しない	入力データ 49 を丸め処理してレジスタ 1 3 1 ~ 1 6 ビットへ出力

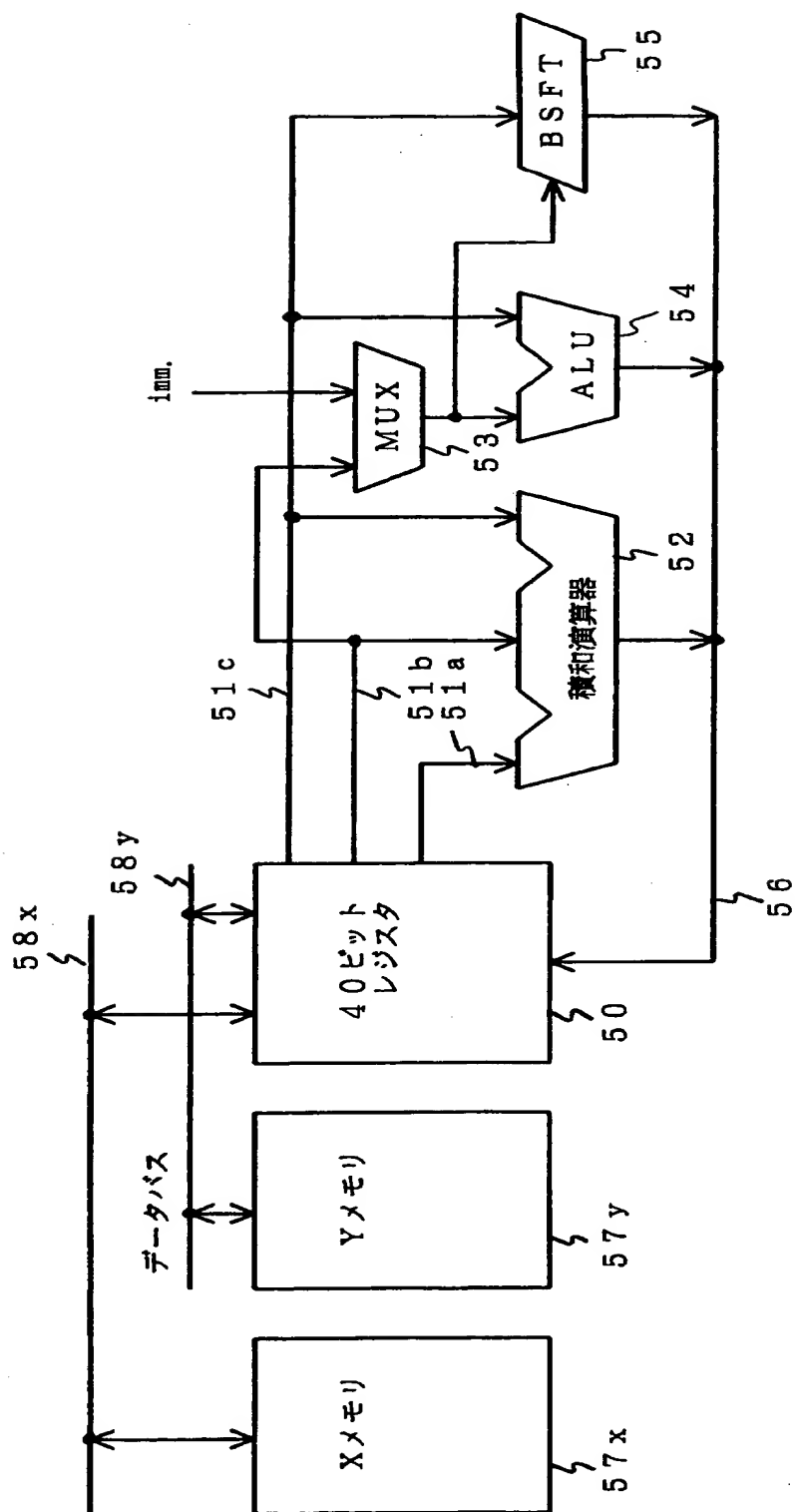
【図 4】



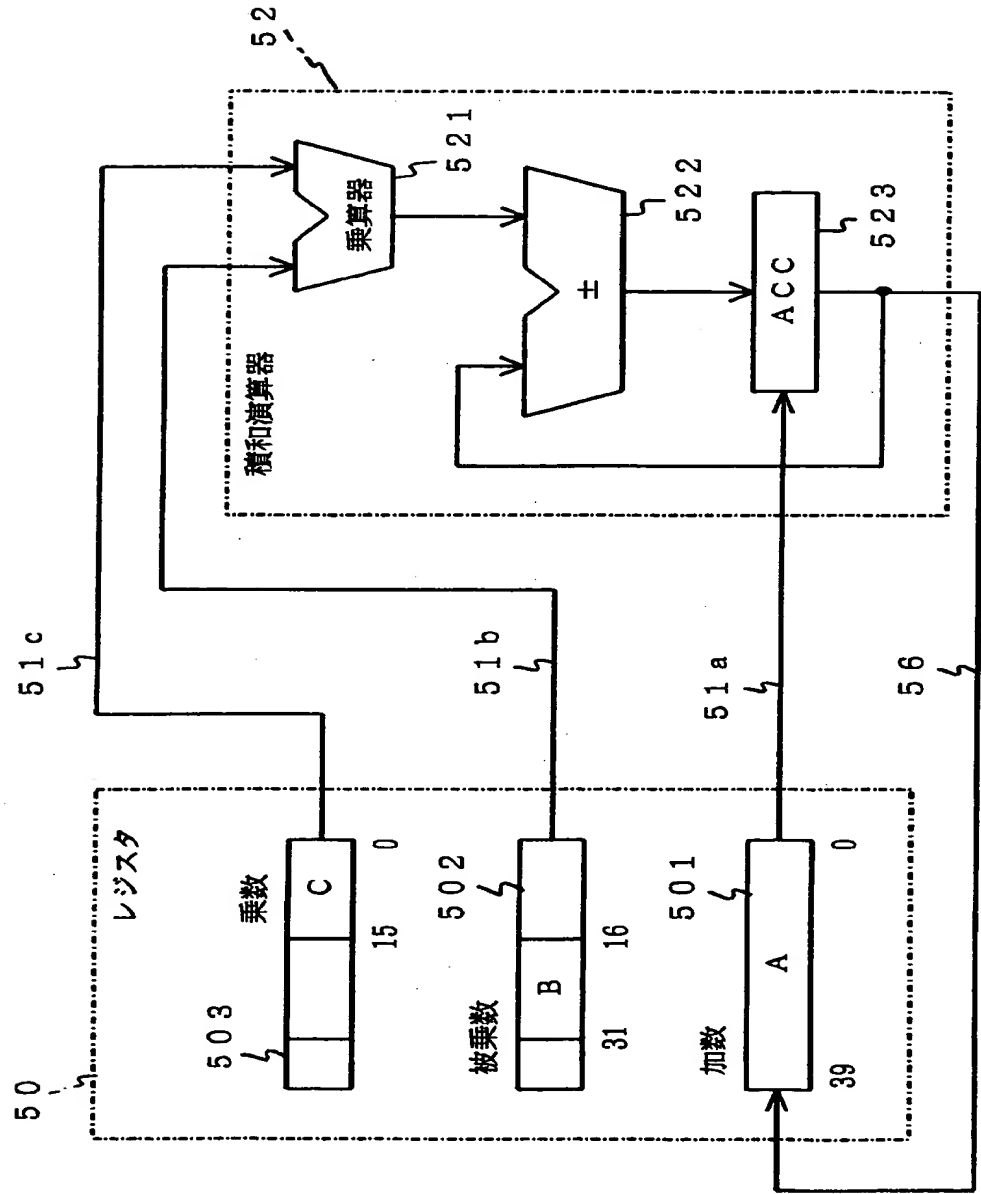
【図 5】



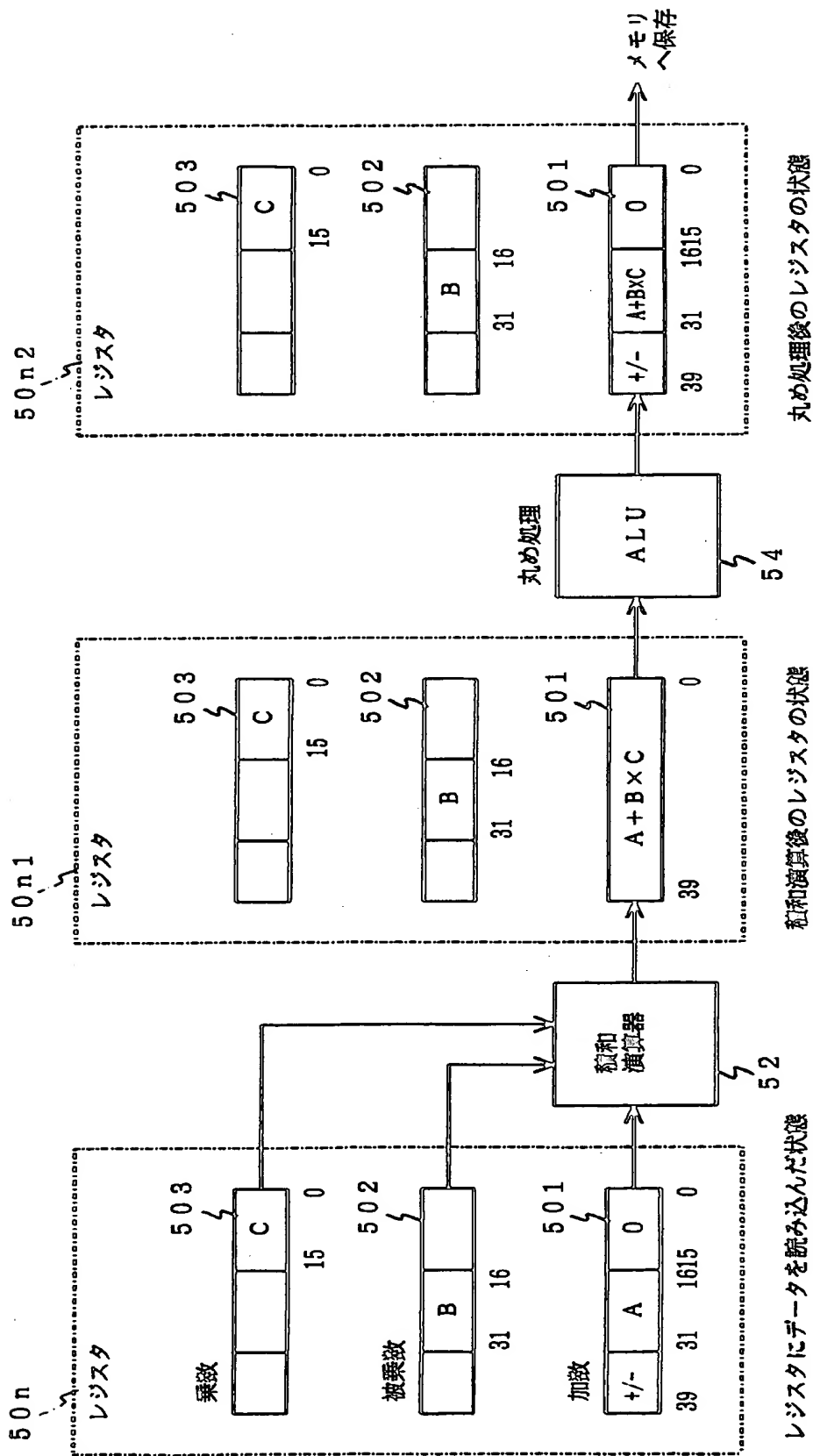
【図 6】



【図 7】



【図 8】



【書類名】 要約書

【要約】

【課題】 レジスタの中での加数の位置を考慮した 1 6 ビットの積和演算を実行可能な丸め機能付き積和演算器を提供する。

【解決手段】 丸め機能付き積和演算器 4 の選択入力及び拡張手段 4 2 は外部からの制御信号 P o s i t i o n が “ 1 ” であれば 4 0 ビットレジスタ 1 の 3 1 ~ 1 6 ビットにあるデータを、制御信号 P o s i t i o n が “ 0 ” であれば 4 0 ビットレジスタ 1 の 1 5 ~ 0 ビットにあるデータをそれぞれ 4 0 ビットに拡張して積和演算器 4 1 へ渡す。積和演算器 4 1 は 4 0 ビットデータと 1 6 ビットデータ被乗数 B と乗数 C とによって積和演算を実行する。丸め処理及び選択出力手段 4 3 は外部からの制御信号 R o u n d が “ 1 ” の場合に 4 0 ビットデータを 1 6 ビットに丸め、制御信号 P o s i t i o n が示す 4 0 ビットレジスタ 1 における加数の位置に丸め処理したデータを出力する。

【選択図】 図 1

特平 11-025674

出 願 人 履 歴 情 報

識別番号 [000004237]

1. 変更年月日 1990年 8月29日

[変更理由] 新規登録

住 所 東京都港区芝五丁目7番1号

氏 名 日本電気株式会社